



# Improving Transliteration with Precise Alignment of Phoneme Chunks and Using Contextual Features

Wei GAO, Kam-Fai WONG and Wai LAM

Department of Systems Engineering & Engineering Management

The Chinese University of Hong Kong

{wgao, kfwong, wlam}@se.cuhk.edu.hk

---

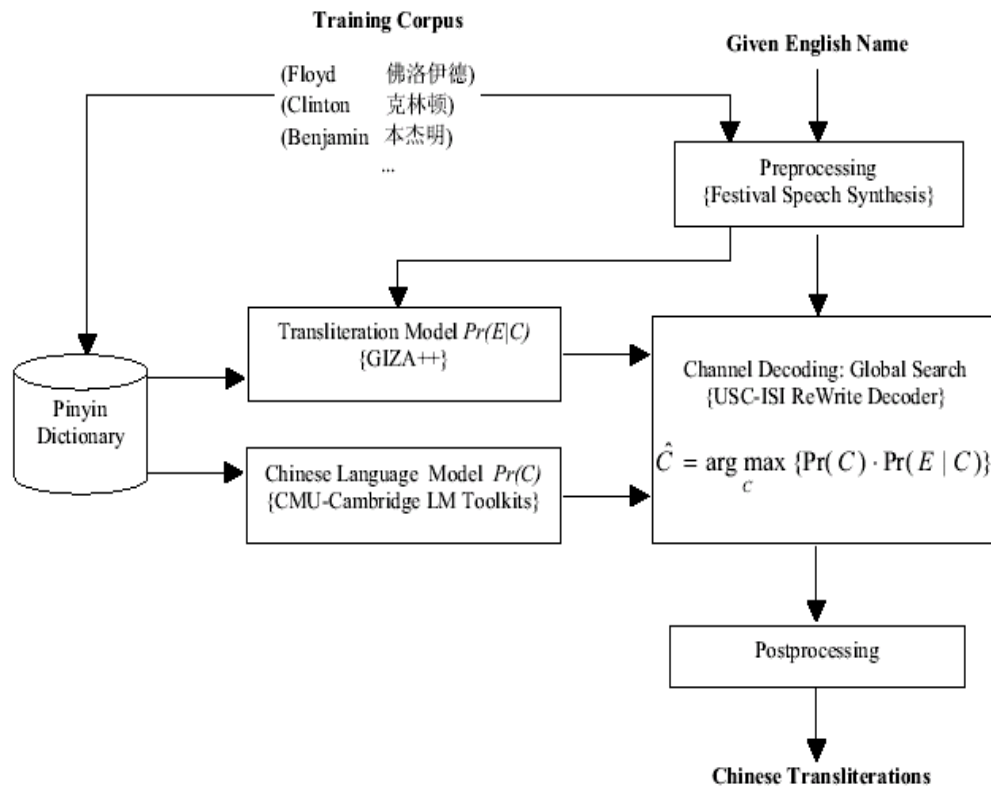
# Outline

- Overview: cross-lingual, OOV and transliteration.
- Related Work: IBM SMT model, source-channel.
- Drawbacks: Why is IBM SMT model limited for transliteration?
- Our Approach: direct transliteration, baseline, improved model.
- Experiments and Discussions
- Conclusions

# Overview

- Automatic transliteration of foreign names (of people, places, organizations, etc) — an important issue in cross language applications. E.g. CLIR, MT, SLP...
- Proper name dictionaries can never be comprehensive rendering name translation ineffective — OOV (Out-Of-Vocabulary) problem.
- How are foreign names translated by people?
  - Rule of thumb
  - Defacto standard, no absolute standard
  - Dialect broadcast: Bin Laden — 本拉登/宾拉登/本拉丹/宾拉丹  
ben la deng    bin la deng    ben la dan    bin la dan
- Machine Transliteration:
  - forward/backward;
  - phoneme-based/grapheme-based;
  - rule-based/statistical

# Related Work—IBM's SMT model based on source-channel (Virga and Khudanpur, 2003, *ACL workshop for NER*)

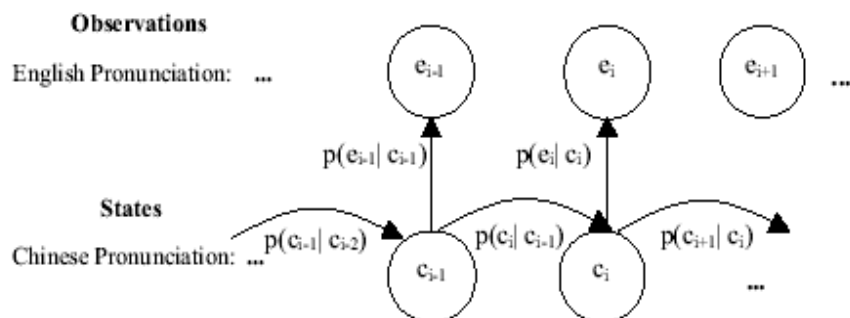


English-to-Chinese transliteration system based on IBM machine translation model described in (Virga and Khudanpur, 2003)

- IBM's SMT model based on Bayes rule:
 
$$\hat{C} = \arg \max_C p(C | E) = \arg \max_C p(E | C) p(C)$$
- $p(E|C)$ : Bootstrapping—EM iterations of Model-1+Model-2+Model-HMM+Model-4
- $$p(E | C) = \sum_A p(E, A | C) \approx p(E, \hat{A} | C)$$
- $p(C)$ : pinyin symbol trigram + Good-Turing smoothing + Katz back-off
- $$p(C) = \prod_{i=1}^{|C|} p(c_i | c_{i-2} c_{i-1})$$
- ReWrite decoder: stack-based algorithm
- Error rates: (Virga and Khudanpur, 2003)

Systems	Training Size	Test Size	Pinyin Errors
Small MT	2233	1541	50.8%
Big MT	3625	250	49.1%

# Drawbacks: Problem 1



- **Problem 1:** Transliteration model  $p(E|C)$  is approximated using zero-order or first-order Markov assumption on state transition and conditional independence assumption between states and observations.
- **Consequence:** With the assumptions, it is hard to extend the model with additional dependencies, such as neighboring phonemes on both sides.

- Model-1: **zero-order** dependency, **one-to-one** alignment model, only uses t table, assuming a uniform alignment probability:

$$p(E, A | C) = \frac{1}{(l+1)^m} \prod_{j=1}^m t(e_j | c_{a_j})$$

- Model-2: **zero-order** dependency, **one-to-one** HMM alignment model, uses t table and alignment probability  $d(a_j | j, l, m)$ :

$$p(E, A | C) = \prod_{j=1}^m t(e_j | c_{a_j}) a(a_j | j, l, m)$$

- Model-HMM: **first-order** dependency, **one-to-one** alignment model, uses t table, assumes alignment position  $a_j$  depends only on its previous alignment position  $a_{j-1}$ :

$$p(E, A | C) = \prod_{j=1}^m t(e_j | c_{a_j}) p(a_j | a_{j-1}, l, m)$$

- Model-4: **zero-order** dependency, **many-to-one** alignment model, uses t, d table, introducing zero-fertility and NULL-generated:

$$p(E, A | C) = \prod_{i=1}^l n(\phi_i | c_i) \prod_{i=0}^l \phi_i! \times$$

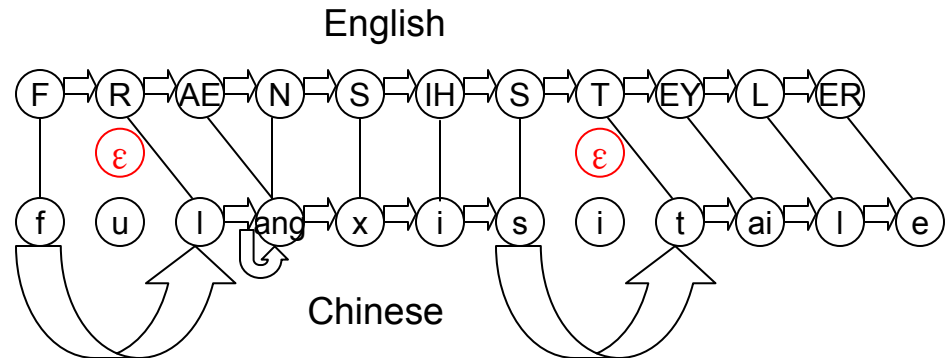
$$C_{\phi_0}^{m-\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times \sum_{j=1}^m t(e_j | c_{a_j}) \times$$

$$\frac{1}{\phi_0!} \times \prod_{j: a_j \neq 0}^m d(j | a_j, l, m)$$

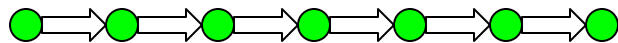
# Drawbacks: Problem 2

English Name	FRANCES TAYLOR										
English Phonemes	F	R	AE	N	S	IHS	TEY	L	ER		
Initials and <u>Finals</u>	f	u	l	ang	x	i	s	t	ai	l	e
Chinese Pinyin	fu	lang	xi	si	tai	le					
Chinese Transliteration	弗	朗	西	丝	泰	勒					

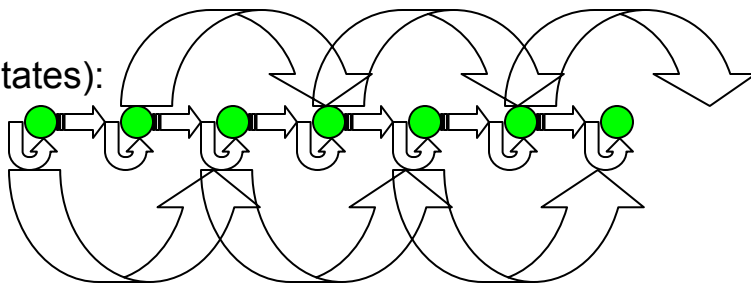
Figure 1. English-to-Chinese transliteration example (Virga and Khudanpur, 2003)



Source (observations):



Target (states):



- Problem 2:** Because the estimation of  $p(E|C)$  is in the opposite direction of E-to-C, it allows only **many-to-one** mapping from source to target.

**Consequence:** Zero fertility symbols /u/, /i/ are spuriously generated from nowhere. They are “deleted” by source-channel during training, but need to be “reproduced” randomly in decoder: zero fertility symbols found {i, e, u, o, ou, ie, ...} are often finals attached to their syllable initials. The stochastic model is less capable of predicting these “disciplinary” zero-fertility finals

---

## Drawbacks: Problem 3

- **Problem 3:** Due to smoothing, language model accepts illegal pinyin sequences that is unobserved in the training data. E.g. two consecutive initials like /kl/. Need to be corrected ad hoc by inserting finals to form legitimate syllables. Wrong transliterations are produced since the insertion of finals has no probabilistic basis.

**Reason:** The transliteration model can wrongly predict zero-fertility pinyin symbols. Example: Clinton → ke lin dun

/K L IH T IN N/ → /k l in dun/ without /e/

# Direct Transliteration Model — Baseline

- Enlightenment from Source-channel:

$$\hat{C} = \arg \max_c p(C | E) = \arg \max_c p(E | C) p(C)$$

$$p(E | C) = \sum_A p(E, A | C) \approx p(E, \hat{A} | C) \rightarrow \text{Viterbi alignment}$$

$$p(C | E) = \sum_A p(C, A | E) \approx p(C, \hat{A} | E)$$

Basic translation model based on Model-3 (Berger et al, 1996, *Computational Linguistics*)

$$p(C, \hat{A} | E) = \prod_{i=1}^{|E|} p(n(e_i) | e_i) \times \prod_{j=1}^{|C|} p(c_j | e_{a_j}) \times d(\hat{A} | E, C)$$

Fertility
Translation
Distortion

Viterbi alignment can be obtained by GIZA++ training by reversing E and C

English Name	FRANCES TAYLOR							
English Phonemes	F	R	AE	N	SIH	S	TEY	LER
Initials and Finals	↓	↓	↓	↓	↓	↓	↓	↓
Chinese Pinyin	fu	lang	xi	si	tai	le		
Chinese Transliteration	弗	朗	西	丝	泰	勒		

$$p(C, \hat{A} | E) \approx \prod_{i=1}^{|E|} p(cmu_i | e_i)$$

*cmu* is individual initial, final or their cluster

## Direct Transliteration Model — Baseline (cont')

- A better approximation:

$$p(C, \hat{A} | E) \approx \prod_{i=1}^{|E|} p(\text{cmu}_i | h_i)$$

$h_i$  is the history or context of  $e_i$ , which can be defined as follows:

$$h_i = \{e_i, e_{i+1}, e_{i+2}, e_{i-1}, e_{i-2}, \text{cmu}_{i-1}, \text{cmu}_{i-2}\}$$

- Estimation Based on MaxEnt Formalism:

$$p(\text{cmu} | h) = \frac{p(h, \text{cmu})}{\sum_{\text{cmu}' \in \Omega} p(h, \text{cmu}')}$$

← Estimate model  $p_\lambda(h, \text{cmu})$

By introducing features  $\{f_1, f_2, \dots, f_m\}$  and associated parameters  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  to express observed events  $(h, \text{cmu})$ ,  $p_\lambda$  is the unique model that maximizes the entropy of the distribution:

$$p_\lambda = \arg \max_p H(p) \quad \text{where} \quad H(p) \equiv - \sum_{h, \text{cmu}} \{p(h, \text{cmu}) \log p(h, \text{cmu})\}$$

under the constraint:  $E(f_i) = \tilde{E}(f_i), 1 \leq i \leq m$

# Direct Transliteration Model — Baseline (cont')

$p_\lambda$  that has the form:  $p(h, cmu) = \mu \prod_{j=1}^m \lambda_j^{f_j(h, cmu)}$

satisfying the constraint maximizes the entropy, where  $\{\mu, \lambda_1, \lambda_2, \dots, \lambda_m\}$  are model parameters and  $\{f_1, f_2, \dots, f_m\}$  are binary feature functions. The model can be obtained by *GIS* (Generalized Iterative Scaling) algorithm (Darroch and Ratcliff, 1972, *Annals of Math. Statistics*; Ratnaparkhi, *EMNLP96*)

Feature selection:  $\hat{f} = \arg \max_f \Delta L(S, f) \equiv \arg \max_f \{L(p_{S \cup f}) - L(p_S)\}$

## ■ Feature template for Transliteration

Category	Contextual Feature Templates	# of Possible Features
1	$e_i = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_E  \cdot  \mathcal{V}_C $
2	$cmu_{i-1} = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_C ^2$
3	$cmu_{i-2}cmu_{i-1} = \mathcal{X}\mathcal{Y}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_C ^3$
4	$e_{i-1} = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_E  \cdot  \mathcal{V}_C $
5	$e_{i-2} = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_E  \cdot  \mathcal{V}_C $
6	$e_{i+1} = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_E  \cdot  \mathcal{V}_C $
7	$e_{i+2} = \mathcal{X}$ and $cmu_i = \mathcal{Z}$	$ \mathcal{V}_E  \cdot  \mathcal{V}_C $

## ■ Example: Extracted features

Position	1	2	3	4	5	6	7	8	9	10	11
English	F	R	AE	N	S	IH	S	T	EY	L	ER
Chinese	fu	l	ang	ε	x	i	si	t	ai	l	e



	Feature Contexts	Feature Predictions
$feature_1$ :	$e_i = /S/$	and $cmu_i = /si/$
$feature_2$ :	$e_{i-1} = /IH/$	and $cmu_i = /si/$
$feature_3$ :	$cmu_{i-1} = /i/$	and $cmu_i = /si/$
$feature_4$ :	$e_{i-2} = /S/$	and $cmu_i = /si/$
$feature_5$ :	$cmu_{i-2} = /x/$ and $cmu_{i-1} = /i/$	and $cmu_i = /si/$
$feature_6$ :	$e_{i+1} = /T/$	and $cmu_i = /si/$
$feature_7$ :	$e_{i+2} = /EY/$	and $cmu_i = /si/$

# Baseline Model Training

**Input:** Raw training instance pairs without alignment

**Output:** features extracted  $\{f_1, f_2, \dots, f_m\}$  and model parameters  $\{\mu, \lambda_1, \lambda_2, \dots, \lambda_m\}$

1. Using EM iterations in GIZA++ to obtain Viterbi alignment. Direction is E-to-C as opposed to C-to-E in baseline. Bootstrapping: 5 model-1 iterations  $\rightarrow$  5 model-2  $\rightarrow$  10 model-HMM  $\rightarrow$  10 model-4;
2. Aligned training instances are passed to *G/S* algorithm in (Ratnaparkhi, *EMNLP96*) to training MaxEnt models. # of iterations = 30.

## Deficiency-1:

Unfavorable clusters:

e.g. Final-initial cluster,

F  $\rightarrow$  /f/ + /R/  $\rightarrow$  /ul/

English Phoneme	Pinyin <i>cmus</i>
AA	iang ch aw ao an uan ai aa eng ong ie ve ia w u o uo e ue ...
AE	ao an ai ab ie ia o uo ui a ε ian ei ang ...
AH	iang axue ou ao an uan aitian ai eng ata aotian ong iu ie ...
AO	ou aw ao an ai aie ong ie ia w u o uo i f e ue a ua iao ε ...
AW	uow ou uok uoh uof of aw ao hao an af ab azhsheng ash iu w u ...
AY	iaai jing uoy wei ay ar an hai ai uai ah uaiy aiy aiai aii aih rong ...

## Deficiency-2:

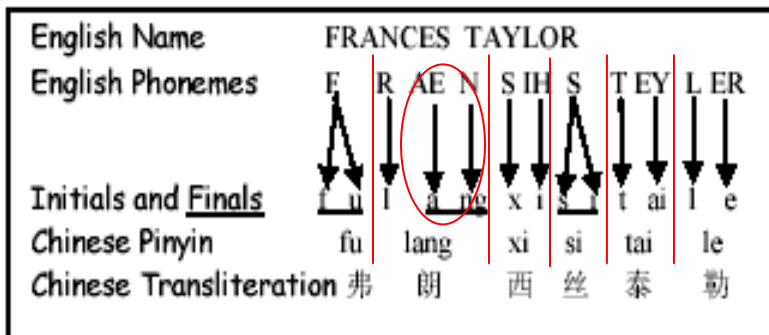
Not allow for many-to-one:

/AE N/  $\rightarrow$  /ang/

Position	1	2	3	4	5	6	7	8	9	10	11
English	F	R	AE	N	S	IH	S	T	EY	L	ER
Chinese	fu	l	ang	ε	x	i	si	t	ai	l	e

# Improving Baseline Model

- Refinement solutions: Precise alignment of phoneme chunks
  - **Solution-1:** To avoid unfavorable clusters, replace GIZA++ training with the *EM* training based on phoneme chunks to prohibit mappings across chunks;
  - **Solution-2:** Decompose compound pinyin finals into basic finals, to reduce granularity and allow for many-to-one from multiple English phonemes to single compound pinyin finals. /AE N/ → /ang/ → /a ng/



Compound Finals (12)	ang	eng	iao	ian	iang	ing
	iong	uai	uan	uang	ong	üan

Basic Finals (24)	a	o	e	ai	ei	ao	ou	er
	an	en	ng	i	ia	ie	iu	in
	u	ua	uo	ui	un	ü	üe	ün

# Alignment of Phoneme Chunks

## Definitions:

- Alignment indicators: A set of indicative units in training sound sequences.
  - For an English phoneme sequence:
    - All the consonants;
    - Vowel at the first position;
    - The second vowel of two contiguous vowels.
  - For a pinyin symbol sequence:
    - All the initials;
    - Final at the first position;
    - The second final of two contiguous finals.
  - Similar set of indicators exist in other Chinese Romanization systems. So they are independent of alignment model.
- Related variables:
  - $\tau(S)$  = # of indicators in a sequence  $S$ ,  $S \in \{E, C\}$ ;
  - $t = \max\{\tau(E), \tau(C)\}$ , represents the maximum # of indicators in  $E$  and  $C$ ;
  - $d = |\tau(E) - \tau(C)|$ , is the difference between the number of indicators in  $E$  and  $C$ .

## Chunking Algorithm:

1. Chunk  $E$  and  $C$  by tagging indicators identified;
2. Compensate the one with fewer indicators by inserting  $d$  mute  $\varepsilon$  at its  $\min\{\tau(E), \tau(C)\}$  possible positions ahead of existing indicators;  $\varepsilon$  is considered as an indicator, so  $\tau(E') = \tau(C) = t$

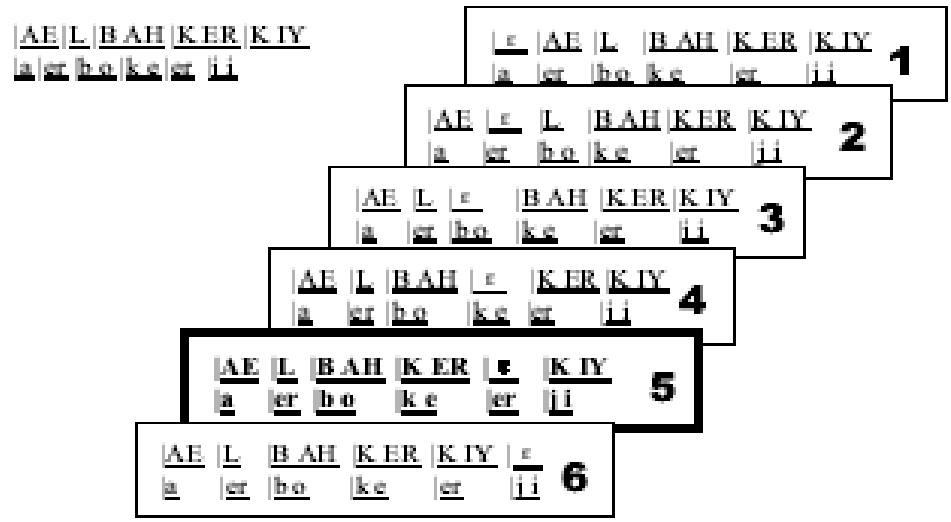
# Alignment of Phoneme Chunks(cont')

## Chunking Algorithm (cont')

3. The  $t$  chunks separated by indicators in  $E'$  should align to the corresponding  $t$  chunks in  $C'$  in the same order;
4. Obtain  $\|A\| = C_t^d = \binom{P_t^d}{d!} = \frac{t!}{(t-d)! d!}$  number of possible chunk alignments with respect to different positions of  $\varepsilon$ .
5. Chunk-level processing: The method can guarantee each chunk contains two sound units at most. In a pair of aligned chunks, only three mapping layouts between elements are possible:
  - e-to-c1c2: c1c2 is considered as a cluster (initial-final);
  - e1e2-to-c1c2: e1-to-c1 and e2-to-c2;
  - e1e2-to-c: Add an additional  $\varepsilon$  at C side; e1-to-c and e2-to- $\varepsilon$  or e1-to- $\varepsilon$  and e2-to-c.  $\|A\| = \|A\| + 1$ ;

### Chunk Example:

(Albuquerque)  
 AE L B AH K ER K IY  
 a er b o k e er j i  
 (阿尔伯克尔基)



# EM Training

## EM Algorithm (Knight and Graehl, ACL97):

1). Initialization: For each English-Chinese pair, assign equal weights to all alignments based on phoneme chunks generated as  $|A|^{-1}$ .

2). Expectation Step: For each of the 40 English phonemes (including  $\epsilon$ ), count the instances of its different mappings from the observations on all alignments produced. Each alignment contributes counts in proportion to its own weight. Normalized the scores the mapping units.

3). Maximization Step: Re-compute the alignment scores. Each alignment is scored with the product of the scores of the symbol-mappings it contains. Normalize the alignment scores.

4). Repeat step 2-3 until the symbol-mapping probabilities converge, meaning that the variation of each probability between two iterations becomes less than a specified threshold.

A1	A2	A3	A4
0.25	0.25	0.25	0.25

e	c	p(c e)	c	p(c e)	c	p(c e)	c	p(c e)
AA	ao	0.625	a	0.272	o	0.100	e	0.003
AE	a	0.519	ai	0.362	ao	0.143		
...	...		...		...		...	

<b>A1</b>	A2	A3	A4
0.54	0.13	0.11	0.22

e	c	p(c e)	c	p(c e)	c	p(c e)	c	p(c e)
AA	ao	0.667	a	0.210	o	0.008	e	0.003
AE	a	0.541	ai	0.322	ao	0.173		
...	...		...		...		...	

<b>A1</b>	A2	A3	A4
0.89	0.04	0.01	0.06

.....

# EM Training (cont')

## Improvement:

Less but finer *cmus*;

e.g. Initials, finals, legal initial-final clusters

English Phoneme	Pinyin <i>cmus</i>
AA	du ou luo yue ao ai ng mi lo le la ya xi sha wo we wa chi ie ...
AE	ao ai ng la ya wa ie ia u o uo lao i ui e a i ao ε ei ...
AH	sai da kai pi lie ou luo wei yue qia ao ai ni hai ng na ...
AO	xiao ou luo yue ao ai yu lo la wo we wa ie ia u o uo huo lao i ...
AW	ou luo bi ao ai wu you iu w u hu o uo i e a i ao ε ei ...
AY	da ci pa wei yue ba ao ai li yi ye ji iu ie ia v u ho o uo l i ...
B	ch bu bo wei bi ba bei ng bai y w s p n l g f e b a fu ε er ...
CH	du di de ci ch bo nu ng na zh ze yi le ye ke xia ji chu z y hu ...

## ■ Decoding Algorithm: Beam search

(Ratnaparkhi, *EMNLP96*)

$$p(\text{cmu}_1^n | e_1^n) \cong \prod_{i=1}^n p(\text{cmu}_i | h_i)$$

$$p(\text{cmu}_i | h_i) = \frac{p(h_i, \text{cmu}_i)}{\sum_{\text{cmu}_i' \in \Omega} p(h_i, \text{cmu}_i')}$$

	Partial history				Beam size	
	1	2	3	...	...	N
e1	x	x	x	x	x	x
e2	x	x	x	x	x	x
e3	x	x	x	x	x	x
e4	?	?	?	?	?	?
...						
...						
en						

# Data Set

- English: ARPABET (IPA in ASCII):

24 consonants	P	T	K	B	D	G	M	N	NG	F	V	TH
	DH	S	Z	SH	ZH	CH	JH	L	W	R	Y	HH
16 vowels	IY	IH	EY	EH	AE	ER	UH	AX	AY	AW	AA	OW
	OY	AO	UW	UH								

- Chinese: Pinyin (Initials & Finals)

23 initials	b	p	m	f	d	t	n	l	g	k	h	j
	q	x	zh	ch	sh	r	z	c	s	y	w	
35 finals	a	o	e	ai	ei	ao	ou	er	an	en	ang	eng
	i	ia	ie	iao	iu	ian	in	iang	ing	iong	u	ua
	uo	uai	ui	uan	un	uang	ong	v	ve	van	vn	

- Data Set

Corpus Used	# of name pairs	Training Size	Close Test Size	Open Test Size
LDC C-E/E-C named entity list v1.0 CMU pronunciation dictionary LDC Hanzi-pinyin conversion table	46,305	41,674 (90%)	4,631 (10%)	4,631 (10%)

# Performance Measures (Kang & Kim, COLING00)

- Character-level Accuracy (C.A.)

$$C.A. = \frac{L - (i + d + s)}{L}$$

→ Edit distance from machine translit. and standard  
→ Length of standard transliteration

- Word-level Accuracy (W.A.)

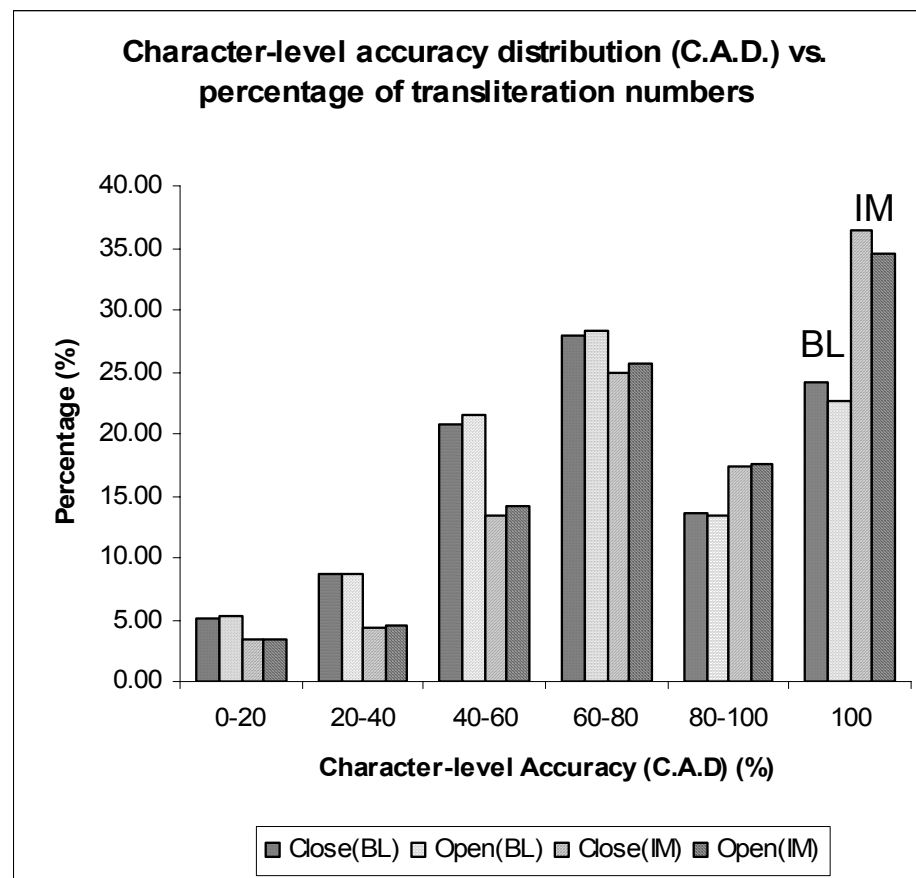
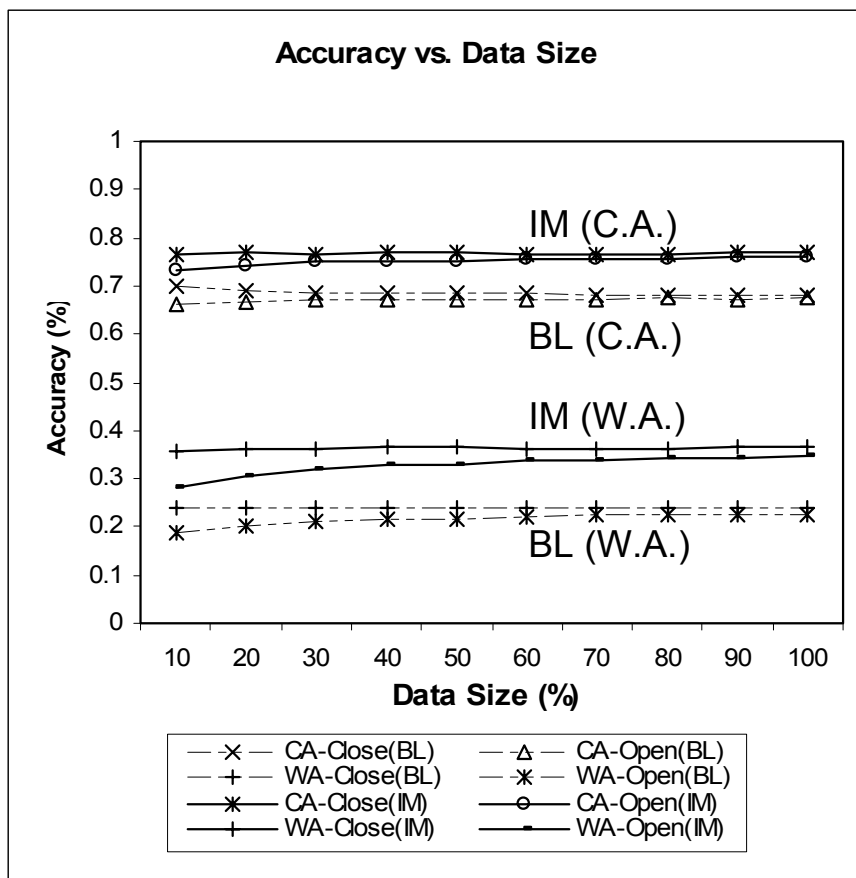
$$W.A. = \frac{\text{\# of correctly transliterated names}}{\text{\# of tested names}}$$

- C.A. Distribution (C.A.D.)

$$C.A.D. = \frac{\text{\# of names with } C.A. \in [r_1, r_2)}{\text{\# of tested names}}$$

→ [0%~20%), [20%~40%), [40%~60%),  
[40%~80%), [80%~100%), [100%]

# Experiment Results — Baseline (BL) vs. Improved (IM)



# Experiment Results — Baseline (BL), Improved (IM) vs. Source-Channel (SC)

Systems		Source-Channel (SC)	Baseline (BL)	Improved (IM)
C.A.	Close	66.35%	68.18%	76.97%
	Open	65.15%	67.18%	75.08%
W.A.	Close	20.73%	23.47%	36.19%
	Open	18.27%	21.49%	32.50%

Original Name	Pronunciation	Machine Trans.	Standard Trans.	C.A.(%)
Voth	V AA TH	wa si	fu te ( 福特 )	0.000
Honcho	HH AO N CH OW	heng huo	ben die ( 本蝶 )	16.67
Pace	P EY S	pei qie	pa cai ( 帕采 )	20.00
Rieth	R AY AH TH	li ao si	li te ( 里特 )	0.000
Fujitsu	F UW JH IH T S UW	fu ji ce	ge jin ( 葛津 )	20.00
Sag	S AE G	sa ge	sa ( 萨 )	0.000
Tokunaga	T OW K UW N AA G AH	tuo ke na jia	de chang ( 德长 )	0.000
Yoho	Y OW HH OW	yue huo	rong feng ( 蓉峰 )	0.000
Hironasa	HH IH R OW M AA S AH	xi luo ma sa	bo ya ( 博雅 )	0.000
Gyocai	G Y OW S EY	ge luo sa	yu cai ( 鱼菜 )	0.000
Bag	B AE G	ba ge	ba ( 巴 )	0.000
Thyme	TH AY M	sai mu	di mei ( 蒂梅 )	20.00
Upshur	AH P SH ER	a pu xiao	e pu she ( 厄普舍 )	16.67
Haim	HH AY M	hai mu	an ( 安 )	0.000
Pet	P EH T	P EH T	bei ( 贝 )	0.000
Shaefer	SH EY F ER	sha fe	xie fu ( 谢弗 )	20.00
Hayer	HH EY ER	hai er	a ye ( 阿耶 )	0.000
Motyka	M AA T AY K AH	ma tai xi	mo di ka ( 莫蒂卡 )	16.67
Flex	F L EH K S	fu lai ke si	fu lai ( 弗莱 )	20.00
Yap	Y AE P	ya pei	ru ( 入 )	0.000
...	...	...	...	...

Table 5.13: Randomly selected sample transliterations with  $C.A. \leq 20\%$

Samples: 100

# of names that are not phonetically translated: 68

e.g. Japanese, Korean or other South Asians' names

Machine transliterations which are phonetically closer to English pronunciations are possibly incorrect. Why?

1. Irregular rule basis by human translators
2. Transliterations based on the pronunciations of their original language. E.g. John, Paris...

**To classify foreign names by their origins**

---

# Conclusions

- Contributions:
  - Identified major deficiencies of SC-based MT model for E-to-C transliteration (3 Problems);
  - Baseline: a one-to-many alignment model with context consideration; without a separate language model, more accurate than SC;
  - Improved model: a refined one-to-many alignment model by precise alignment of phoneme chunks and allowing for many-to-one using smaller granularity phonetic representations. Significant improvement.
- Future work:
  - Incorporate different/additional context, language model, composition of direct and inverted model.
  - Classify foreign names by their origins and translate in terms of original pronunciation.
  - Support many-to-many symbolic mapping.

---

Thank you!

---